



# Ising-Based Consensus Clustering on Specialized Hardware

Eldan Cohen<sup>1</sup>(✉), Avradip Mandal<sup>2</sup>, Hayato Ushijima-Mwesigwa<sup>2</sup>,  
and Arnab Roy<sup>2</sup>

<sup>1</sup> University of Toronto, Toronto, Canada  
ecohen@mie.utoronto.ca

<sup>2</sup> Fujitsu Laboratories of America, Inc., Sunnyvale, USA  
{amandal,hayato,aroy}@us.fujitsu.com

**Abstract.** The emergence of specialized optimization hardware such as CMOS annealers and adiabatic quantum computers carries the promise of solving hard combinatorial optimization problems more efficiently in hardware. Recent work has focused on formulating different combinatorial optimization problems as Ising models, the core mathematical abstraction used by a large number of these hardware platforms, and evaluating the performance of these models when solved on specialized hardware. An interesting area of application is data mining, where combinatorial optimization problems underlie many core tasks. In this work, we focus on consensus clustering (clustering aggregation), an important combinatorial problem that has received much attention over the last two decades. We present two Ising models for consensus clustering and evaluate them using the Fujitsu Digital Annealer, a quantum-inspired CMOS annealer. Our empirical evaluation shows that our approach outperforms existing techniques and is a promising direction for future research.

## 1 Introduction

The increasingly challenging task of scaling the traditional Central Processing Unit (CPU) has led to the exploration of new computational platforms such as quantum computers, CMOS annealers, neuromorphic computers, and so on (see [3] for a detailed exposition). Although their physical implementations differ significantly, adiabatic quantum computers, CMOS annealers, memristive circuits, and optical parametric oscillators all share Ising models as their core mathematical abstraction [3]. This has led to a growing interest in the formulation of computational problems as Ising models and in the empirical evaluation of these models on such novel computational platforms. This body of literature includes clustering and community detection [14, 19, 23], graph partitioning [26], and many NP-Complete problems such as covering, packing, and coloring [17].

Consensus clustering is the problem of combining multiple ‘base clusterings’ of the same set of data points into a single consolidated clustering [9]. Consensus clustering is used to generate robust, stable, and more accurate clustering

---

E. Cohen—Work done while at Fujitsu Laboratories of America.

© The Author(s) 2020

M. R. Berthold et al. (Eds.): IDA 2020, LNCS 12080, pp. 106–118, 2020.

[https://doi.org/10.1007/978-3-030-44584-3\\_9](https://doi.org/10.1007/978-3-030-44584-3_9)

results compared to a single clustering approach [9]. The problem of consensus clustering has received significant attention over the last two decades [9], and was previously considered under different names (clustering aggregation, cluster ensembles, clustering combination) [10]. It has applications in different fields including data mining, pattern recognition, and bioinformatics [10] and a number of algorithmic approaches have been used to solve this problem. The consensus clustering is, in essence, a combinatorial optimization problem [28] and different instances of the problem have been proven to be NP-hard (e.g., [6, 25]).

In this work, we investigate the use of special purpose hardware to solve the problem of consensus clustering. To this end, we formulate the problem of consensus clustering using Ising models and evaluate our approach on a specialized CMOS annealer. We make the following contributions:

1. We present and study two Ising models for consensus clustering that can be solved on a variety of special purpose hardware platforms.
2. We demonstrate how our models are embedded on the Fujitsu Digital Annealer (DA), a quantum-inspired specialized CMOS hardware.
3. We present an empirical evaluation based on seven benchmark datasets and show our approach outperforms existing techniques for consensus clustering.

## 2 Background

### 2.1 Problem Definition

Let  $X = \{x_1, \dots, x_n\}$  be a set of  $n$  data points. A *clustering* of  $X$  is a process that partitions  $X$  into subsets, referred to as *clusters*, that together cover  $X$ . A clustering is represented by the mapping  $\pi : X \rightarrow \{1, \dots, k_\pi\}$  where  $k_\pi$  is the number of clusters produced by clustering  $\pi$ . Given  $X$  and a set  $\Pi = \{\pi_1, \dots, \pi_m\}$  of  $m$  clusterings of the points in  $X$ , the *Consensus Clustering Problem* is to find a new clustering,  $\pi^*$ , of the data  $X$  that best summarizes the set of clusterings  $\Pi$ . The new clustering  $\pi^*$  is referred to as the *consensus clustering*.

Due to the ambiguity in the definition of an optimal consensus clustering, several approaches have been proposed to measure the solution quality of consensus clustering algorithms [9]. In this work, we focus on the approach of determining a consensus clustering that agrees the most with the original clusterings. As an objective measure to determine this agreement, we use the mean Adjusted Rand Index (ARI) metric (Eq. 14). However, we also consider clustering quality measured by mean Silhouette Coefficient [22] and clustering accuracy based on true labels. In Sect. 4 these evaluation criteria are discussed in more details.

### 2.2 Existing Criteria and Methods

Various criteria or objectives have been proposed for the Consensus Clustering Problem. In this work we mainly focus on two well-studied criteria, one based on the pairwise similarity of the data points, and the other based on the different assignments of the base clusterings. Other well-known criteria and objectives for the Consensus Clustering Problem can be found in the excellent surveys of [9, 27], with most defining NP-Hard optimization problems.

*Pairwise Similarity Approaches:* In this approach, a similarity matrix  $S$  is constructed such that each entry in  $S$  represents the fraction of clusterings in which two data points belong to the same cluster [20]. In particular,

$$S_{uv} = \frac{1}{m} \sum_{i=1}^m \mathbb{1}(\pi_i(u) = \pi_i(v)), \quad (1)$$

with  $\mathbb{1}$  being the indicator function. The value  $S_{uv}$  lies between 0 and 1, and is equal to 1 if all the base clusterings assign points  $u$  and  $v$  to the same cluster. Once the pairwise similarity matrix is constructed, one can use any similarity-based clustering algorithm on  $S$  to find a consensus clustering with a fixed number of clusters,  $K$ . For example, [16] proposed to find a consensus clustering  $\pi^*$  with exactly  $K$  clusters that minimizes the within-cluster dissimilarity:

$$\min \sum_{\substack{u,v \in X: \\ \pi^*(u) = \pi^*(v)}} (1 - S_{uv}). \quad (2)$$

*Partition Difference Approaches:* An alternative formulation is based on the different assignments between clustering. Consider two data points  $u, v \in X$ , and two clusterings  $\pi_i, \pi_j \in \Pi$ . The following binary indicator tests if  $\pi_i$  and  $\pi_j$  disagree on the clustering of  $u$  and  $v$ :

$$d_{u,v}(\pi_i, \pi_j) = \begin{cases} 1, & \text{if } \pi_i(u) = \pi_i(v) \text{ and } \pi_j(u) \neq \pi_j(v) \\ 1, & \text{if } \pi_i(u) \neq \pi_i(v) \text{ and } \pi_j(u) = \pi_j(v) \\ 0, & \text{otherwise.} \end{cases} \quad (3)$$

The distance between two clusterings is then defined based on the number of pairwise disagreements:

$$d(\pi_i, \pi_j) = \frac{1}{2} \sum_{u,v \in X} d_{u,v}(\pi_i, \pi_j) \quad (4)$$

with the  $\frac{1}{2}$  factor to take care of double counting and can be ignored. This measure is defined as the number of pairs of points that are in the same cluster in one clustering and in different clusters in the other, essentially considering the (unadjusted) Rand index [9]. Given this measure, a common objective is to find a consensus clustering  $\pi^*$  with respect to the following optimization problem:

$$\min \sum_{i=1}^m d(\pi_i, \pi^*). \quad (5)$$

*Methods and Algorithms:* The two different criteria given above define fundamentally different optimization problems, thus different algorithms have been proposed. One key difference between the two approaches inherently lies in determining the number of clusters  $k_{\pi^*}$  in  $\pi^*$ . The pairwise similarity approaches (e.g.,

Eq. (2)) require an input parameter  $K$  that fixes the number of clusters in  $\pi^*$ , whereas the partition difference approaches such as Eq. (5) do not have this requirement and determining  $k_{\pi^*}$  is part of the objective of the problem. Therefore, for example, Eq. (2) will have a minimum value in the case when  $k_{\pi^*} = n$ , however this does not hold for Eq. (5).

The Cluster-based Similarity Partitioning Algorithm (CSPA) is proposed in [24] for solving the pairwise similarity based approach. The CSPA constructs a similarity-based graph with each edge having a weight proportional to the similarity given by  $S$ . Determining the consensus clustering with exactly  $K$  clusters is treated as a  $K$ -way graph partitioning problem, which is solved by methods such as METIS [12]. In [20], the authors experiment with different clustering algorithms including hierarchical agglomerative clustering (HAC) and iterative techniques that start from an initial partition and iteratively reassign points to clusters based on their pairwise similarities. For the partition difference approach, Li et al. [15] proposed to solve Eq. (5) using nonnegative matrix factorization (NMF). Gionis et al. [10] proposed several algorithms that make use of the connection between Eq. (5) and the problem of correlation clustering. CSPA, HAC, NMF: these three approaches are considered as baseline in our empirical evaluation section (Sect. 4).

### 2.3 Ising Models

Ising models are graphical models that include a set of nodes representing spin variables and a set of edges corresponding to the interactions between the spins. The energy level of an Ising model which we aim to minimize is given by:

$$E(\sigma) = \sum_{(i,j) \in \mathcal{E}} J_{i,j} \sigma_i \sigma_j + \sum_{i \in \mathcal{N}} h_i \sigma_i, \quad (6)$$

where the variables  $\sigma_i \in \{-1, 1\}$  are the spin variables and the couplers,  $J_{i,j}$ , represent the interaction between the spins.

A Quadratic Unconstrained Binary Optimization (QUBO) model includes binary variables  $q_i \in \{0, 1\}$  and couplers,  $c_{i,j}$ . The objective to minimize is:

$$E(\mathbf{q}) = \sum_{i=1}^n c_i q_i + \sum_{i < j} c_{i,j} q_i q_j. \quad (7)$$

QUBO models can be transformed to Ising models by setting  $\sigma_i = 2q_i - 1$  [2].

## 3 Ising Approach for Consensus Clustering on Specialized Hardware

In this section, we present our approach for solving consensus clustering on specialized hardware using Ising models. We present two Ising models that correspond to the two approaches in Sect. 2.2. We then demonstrate how they can be solved on the Fujitsu Digital Annealer (DA), a specialized CMOS hardware.

### 3.1 Pairwise Similarity-Based Ising Model

For each data point  $u \in X$ , let  $q_{uc} \in \{0, 1\}$  be the binary variable such that  $q_{uc} = 1$  if  $\pi^*$  assigns  $u$  to cluster  $c$ , and 0 otherwise. Then the constraints

$$\sum_{c=1}^K q_{uc} = 1, \quad \text{for each } u \in X \quad (8)$$

ensure  $\pi^*$  assigns each point to exactly one cluster. Subject to the constraints (8), the sum of quadratic terms  $\sum_{c=1}^K q_{uc}q_{vc}$  is 1 if  $\pi^*$  assigns both  $u, v \in X$  to the same cluster, and is 0 if assigned to different clusters. Therefore the value

$$\sum_{\substack{u, v \in X: \\ \pi^*(u) = \pi^*(v)}} (1 - S_{uv}) = \sum_{u, v \in X} (1 - S_{uv}) \sum_{c=1}^K q_{uc}q_{vc} \quad (9)$$

represents the sum of within-cluster dissimilarities in  $\pi^*$ :  $(1 - S_{uv})$  is the fraction of clusterings in  $\Pi$  that assign  $u$  and  $v$  to different clusters while  $\pi^*$  assigns them to the same cluster. We therefore reformulate Eq. (2) as QUBO:

$$\min \sum_{u, v \in X} (1 - S_{uv}) \sum_{c=1}^K q_{uc}q_{vc} + \sum_{u \in X} A \left( \sum_{c=1}^K q_{uc} - 1 \right)^2. \quad (10)$$

where the term  $\sum_{u \in X} A \left( \sum_{c=1}^K q_{uc} - 1 \right)^2$  is added to the objective function to ensure that the constraints (8) are satisfied.  $A$  is positive constant that penalizes the objective for violations of constraints (8). One can show that if  $A \geq n$ , the optimal solution of the QUBO in Eq. (10) does not violate the constraints (8). The proof is very similar to proof of Theorem 1 and a similar result in [14].

### 3.2 Partition Difference Ising Model

The partition difference approach essentially considers the (unadjusted) Rand Index [9] and therefore can be expected to perform better. The *Correlation Clustering Problem* is another important problem in data mining. Gionis et al. [10] showed that Eq. (5) is a restricted case of the Correlation Clustering Problem, and that Eq. (5) can be expressed as the following equivalent form of the Correlation Clustering Problem

$$\min_{\pi^*} \sum_{\substack{u, v \in X: \\ \pi^*(u) = \pi^*(v)}} (1 - S_{uv}) + \sum_{\substack{u, v \in X: \\ \pi^*(u) \neq \pi^*(v)}} S_{uv}. \quad (11)$$

We take advantage of this equivalence to model Eq. (5) as a QUBO. In a similar fashion to the QUBO formulated in the preceding subsection, the terms

$$\sum_{\substack{u, v \in X: \\ \pi^*(u) \neq \pi^*(v)}} S_{uv} = \sum_{u, v \in X} S_{uv} \sum_{1 \leq c \neq l \leq K} q_{uc}q_{vl} \quad (12)$$

measure the similarity between points in *different* clusters, where  $K$  represents an *upper bound* for the number of clusters in  $\pi^*$ . This then leads to the minimizing the following QUBO:

$$\sum_{u,v \in X} (1 - S_{uv}) \sum_{c=1}^K q_{uc}q_{vc} + \sum_{u,v \in X} S_{uv} \sum_{1 \leq c \neq l \leq K} q_{uc}q_{vl} + \sum_{u \in X} B \left( \sum_{c=1}^K q_{uc} - 1 \right)^2. \quad (13)$$

Intuitively, Eq. (13) measures the disagreement between the consensus clustering and the clusterings in  $\Pi$ . This disagreement is due to points that are clustered together in the consensus clustering but not in the clusterings in  $\Pi$ , however it is also due to points that are assigned to different clusters in the consensus partition but in the same cluster in some of the partitions in  $\Pi$ .

Formally, we can show that Eq. (13) is equivalent to the correlation clustering formulation in Eq. (11) when setting  $B \geq n$ . Consistent with other methods that optimize Eq. (5) (e.g., [15]), our approach takes as an input  $K$ , an *upper bound* on the number of clusters in  $\pi^*$ , however the obtained solution can use smaller number of clusters. In our proof, we assume  $K$  is large enough to represent the optimal solution, i.e., greater than the number of clusters in optimal solutions to the correlation clustering problem in Eq. (11).

**Theorem 1.** *Let  $\bar{\mathbf{q}}$  be the optimal solution to the QUBO given by Eq. (13). If  $B \geq n$ , for a large enough  $K \leq n$ , an optimal solution to the Correlation Clustering Problem in Eq. (11),  $\bar{\pi}$ , can be efficiently evaluated from  $\bar{\mathbf{q}}$ .*

*Proof.* First we show the optimal solution to the QUBO in Eq. (13) satisfies the one-hot encoding ( $\sum_k q_{uk} = 1$ ). This would imply given  $\bar{\mathbf{q}}$  we can create a valid clustering  $\bar{\pi}$ . Note, the optimal solution will never have  $\sum_c q_{uc} > 1$  as it can only increase the cost. The only case in which an optimal solution will have  $\sum_c q_{uc} < 1$  is when the cost of assigning a point to a cluster is higher than the cost of not assigning it to a cluster (i.e., the penalty  $B$ ). Assigning a point  $u$  to a cluster will incur a cost of  $(1 - S_{uv})$  for each point  $v$  in the same cluster and  $S_{uv}$  for each point  $v$  that is not in the cluster. As there is additional  $n - 1$  points in total, and both  $(1 - S_{uv})$  and  $S_{uv}$  are less or equal to one (Eq. (1)), setting  $B \geq n$  guarantees the optimal solution satisfies the one-hot encoding.

Now we assume that  $\bar{\pi}$  is not optimal, i.e., there exists an optimal solution  $\hat{\pi}$  to Eq. (11) that has a strictly lower cost than  $\bar{\pi}$ . Let  $\hat{\mathbf{q}}$  be the corresponding QUBO solution to  $\hat{\pi}$ , such that  $\bar{\pi}(u) = k$  if and only if  $\hat{q}_{uk} = 1$ . This is possible because  $K$  is large enough to accommodate all clusters in  $\hat{\pi}$ . As both  $\bar{\mathbf{q}}$  and  $\hat{\mathbf{q}}$  satisfy that one-hot encoding (penalty terms are zero), their cost is identical to the cost of  $\bar{\pi}$  and  $\hat{\pi}$ . Since the cost of  $\hat{\pi}$  is strictly lower than  $\bar{\pi}$ , and the cost of  $\hat{\mathbf{q}}$  is lower or equal to  $\bar{\mathbf{q}}$ , we have a contradiction.  $\square$

### 3.3 Solving Consensus Clustering on the Fujitsu Digital Annealer

The Fujitsu Digital Annealer (DA) is a recent CMOS hardware for solving combinatorial optimization problems formulated as QUBO [1, 8]. We use the second

generation of the DA that is capable of representing problems with up to 8192 variables with up to 64 bits of precision. The DA has previously been used to solve problems in areas such as communication [18] and signal processing [21].

The DA algorithm [1] is based on simulated annealing (SA) [13], while taking advantage of the massive parallelization provided by the CMOS hardware [1]. It has several key differences compared to SA, most notably a *parallel-trial* scheme in which each MC step considers all possible one-bit flips in parallel and *dynamic offset* mechanism that increase the energy of a state to escape local minima [1].

**Encoding Consensus Clustering on the DA.** When embedding our Ising models on the DA, we need to consider the hardware specification and adapt the representation of our model accordingly. Due to hardware precision limit, we need to embed the couplers and biases on an integer scale with limited granularity. In our experiments, we normalize the pairwise costs  $S_{uv}$  in the discrete range  $[0, 100]$ ,  $D_{ij} = \lceil S_{uv} \cdot 100 \rceil$ , and accordingly  $(1 - S_{uv})$  is replaced by  $(100 - D_{uv})$ . Note that the theoretical bound  $B = n$  is adjusted accordingly to be  $B = 100 \cdot n$ .

The theoretical bound guarantees that all constraints are satisfied if problems are solved to optimality. In practice, the DA does not necessarily solve problems to optimality and due to the nature of annealing-based algorithms, using very high weights for constraints is likely to create deep local minima and result in solutions that may satisfy the constraints but are often of low-quality. This is especially relevant to our pairwise similarity model where the bound tends to become loose as the number of clusters grows. In our experiments, we use constant, reasonably high, weights that were empirically found to perform well across datasets. For the pairwise similarity-based model (Eq. (10)) we use  $A = 2^{14}$ , and for the partition difference model (Eq. (13)) we use  $B = 2^{15}$ . While we expect to get better performance by tuning the weights per-dataset, our goal is to demonstrate the performance of our approach in a general setting. Automatic tuning of the weight values for the DA is a direction for future work.

Unlike many of the existing consensus clustering algorithms that run until convergence, our method runs for a given time limit (defined by the number of runs and iterations) and returns the best solution encountered. In our experiments, we arbitrarily choose *three seconds* as a (reasonably short) time limit to solve our Ising models. As with the weights, we employ a single temperature schedule across all datasets, and *do not* tune it per dataset.

## 4 Empirical Evaluation

We perform an extensive empirical evaluation of our approach using a set of seven benchmark datasets. We first describe how we generate the set of clusterings, *II*. Next, we describe the baselines, the evaluation metrics, and the datasets.

**Generating Partitions.** We follow [7] and generate a set of clusterings by randomizing the parameters of the K-Means algorithm, namely the number of

clusters  $K$  and the initial cluster centers. In this work, we only use labelled datasets for which we know the number of clusters,  $\tilde{K}$ , based on the true labels. To generate the base clusterings we run the K-Means algorithm with random cluster centers and we randomly choose  $K$  from the range  $[2, 3\tilde{K}]$ . For each dataset, we generate 100 clusterings to serve as the clustering set  $\Pi$ .

**Baseline Algorithms.** We compare our pairwise similarity-based Ising model, referred to as DA-Sm, and our correlation clustering Ising model, referred to as DA-Cr, to three popular algorithms for consensus clustering:

1. The cluster-based similarity partitioning algorithm (CSPA) [24] solved as a  $K$ -way graph partitioning problem using METIS [12].
2. The nonnegative matrix factorization (NMF) formulation in [15].
3. Hierarchical agglomerative clustering (HAC) starts with all points in singleton clusters and repeatedly merges the two clusters with the largest average similarity based on  $S$ , until reaching the desired number of clusters [20].

**Evaluation.** We evaluate the different methods using three measures. Our main concern in this work is the level of agreement between the consensus clustering and the set of input clusterings. To this end, one requires a metric measuring the similarity of two clusterings that can be used to measure how close the consensus clustering  $\pi^*$  to each base clustering  $\pi_i \in \Pi$  is. One of popularly used metrics to measure the similarity between two clusterings is the Rand Index (RI) and Adjusted Rand Index (ARI) [11]. The Rand Index of two clustering lies between 0 and 1, obtaining the value 1 when both clusterings perfectly agree. Likewise, the maximum score of ARI, which is corrected-for-chance version of RI, is achieved when both clusterings perfectly agree.  $ARI(\pi_i, \pi^*)$  can be viewed as measure of *agreement* between the consensus clustering  $\pi^*$  and some base clusterings  $\pi_i \in \Pi$ . We use the mean ARI as the main evaluation criteria:

$$\frac{1}{m} \sum_{i=1}^m ARI(\pi_i, \pi^*) \quad (14)$$

We also evaluate  $\pi^*$  based on clustering quality and accuracy. For clustering quality, we use the mean Silhouette Coefficient [22] of all data points (computed using the Euclidean distance between the data points). For clustering accuracy, we compute the ARI between the consensus partition  $\pi^*$  and the true labels.

**Benchmark Datasets.** We run experiments on seven datasets with different characteristics: *Iris*, *Optdigits*, *Pendigits*, *Seeds*, *Wine* from the UCI repository [5] as well as *Protein* [29] and *MNIST*.<sup>1</sup> *Optdigits-389* is a randomly sampled subset of *Optdigits* containing only the digits  $\{3, 8, 9\}$ . Similarly, *MNIST-3689* and *Pendigits-149* are subsets of the MNIST and *Pendigits* datasets.

<sup>1</sup> <http://yann.lecun.com/exdb/mnist/>.



Table 1 provides statistics on each of the data set, with the coefficient of variation (CV) [4] describing the degree of class imbalance: zero indicates perfectly balanced classes, while higher values indicate higher degree of class imbalance.

**Table 1.** Datasets

Dataset	# Instances	# Features	# Clusters	CV
Iris	150	4	3	0.000
MNIST-3689	389	784	4	0.015
Optdigits-389	537	64	3	0.021
Pendigits-149	532	16	3	0.059
Protein	116	20	6	0.301
Seeds	210	7	3	0.000
Wine	178	13	3	0.158

#### 4.1 Results

We compare the baseline algorithms to the two Ising models in Sect. 3 solved using the Fujitsu Digital Annealer described in Sect. 3.3.

Clustering is typically an unsupervised task and the number of clusters is unknown. The number of clusters in the true labels,  $\tilde{K}$ , is not available in real scenarios. Furthermore,  $\tilde{K}$  is not necessarily the best value for clustering tasks (e.g., in many cases it is better to have smaller clusters that are more pure). We therefore test the algorithms in two configurations: when the number of clusters is set to  $\tilde{K}$ , as in the true labels, and when the number of clusters is set to  $2\tilde{K}$ .

**Table 2.** Consensus performance measured by mean ARI across partitions

Dataset	$\tilde{K}$ clusters					$2\tilde{K}$ clusters				
	CSPA	NMF	HAC	DA-Sm	DA-Cr	CSPA	NMF	HAC	DA-Sm	DA-Cr
Iris	0.555	<b>0.618</b>	<b>0.618</b>	<b>0.619</b>	<b>0.621</b>	0.536	0.614	0.627	0.608	<b>0.642</b>
MNIST	0.459	0.449	0.469	<b>0.474</b>	<b>0.474</b>	0.456	0.511	0.517	0.490	<b>0.521</b>
Optdig.	0.528	<b>0.550</b>	0.541	<b>0.550</b>	<b>0.551</b>	0.492	0.596	0.608	0.576	<b>0.612</b>
Pendig.	0.546	0.546	0.507	<b>0.555</b>	<b>0.555</b>	0.531	0.629	<b>0.642</b>	0.605	<b>0.644</b>
Protein	0.344	0.393	0.379	0.390	<b>0.405</b>	0.324	0.419	<b>0.423</b>	0.378	0.415
Seeds	0.558	<b>0.577</b>	0.534	<b>0.575</b>	<b>0.577</b>	0.484	0.602	0.602	0.580	<b>0.612</b>
Wine	0.481	<b>0.536</b>	0.535	<b>0.537</b>	<b>0.538</b>	0.502	<b>0.641</b>	<b>0.641</b>	<b>0.641</b>	<b>0.643</b>
# Best	0	4	1	6	<b>7</b>	0	1	3	1	<b>6</b>

**Consensus Criteria.** Table 2 shows the mean ARI between  $\pi^*$  and the clusterings in  $\Pi$ . To avoid bias due to very minor differences, we consider all the methods that achieved Mean ARI that is within a threshold of 0.0025 from the best method to be equivalent and highlight them in bold. We also summarize the number of times each method was considered best across the different datasets.

The results show that DA-Cr is the best performing method for both  $\tilde{K}$  and  $2\tilde{K}$  clusters. The results of DA-Sm are not consistent: DA-Sm and NMF are performing well for  $\tilde{K}$  clusters and HAC is performing better for  $2\tilde{K}$  clusters.

**Clustering Quality.** Table 3 report the mean Silhouette Coefficient of all data points. Again, DA-Cr is the best performing method across datasets, followed by HAC. NMF seems to be equivalent to HAC for  $2\tilde{K}$ .

**Table 3.** Clustering quality measured by Silhouette

Dataset	$\tilde{K}$ clusters					$2\tilde{K}$ clusters				
	CSPA	NMF	HAC	DA-Sm	DA-Cr	CSPA	NMF	HAC	DA-Sm	DA-Cr
Iris	0.519	<b>0.555</b>	<b>0.555</b>	0.551	<b>0.553</b>	0.289	0.366	<b>0.371</b>	0.343	<b>0.373</b>
MNIST	0.075	0.072	<b>0.078</b>	<b>0.079</b>	<b>0.078</b>	0.069	<b>0.082</b>	0.074	0.074	<b>0.082</b>
Optdig.	0.127	0.120	0.120	<b>0.130</b>	<b>0.130</b>	0.088	<b>0.119</b>	<b>0.119</b>	0.112	<b>0.121</b>
Pendig.	0.307	0.307	<b>0.315</b>	0.310	0.310	0.305	0.332	<b>0.375</b>	0.368	0.364
Protein	0.074	<b>0.106</b>	0.095	0.094	<b>0.104</b>	0.068	0.111	0.115	<b>0.119</b>	<b>0.118</b>
Seeds	0.461	0.468	0.410	0.469	<b>0.472</b>	0.275	<b>0.343</b>	0.304	<b>0.344</b>	0.302
Wine	0.453	0.542	<b>0.571</b>	0.547	0.545	0.452	<b>0.543</b>	<b>0.541</b>	0.539	<b>0.542</b>
# Best	0	2	4	2	<b>5</b>	0	4	4	2	<b>5</b>

**Clustering Accuracy.** Table 4 shows the clustering accuracy measured by the ARI between  $\pi^*$  and the true labels. For  $\tilde{K}$ , we find DA-Sm to be best-performing solution (followed by DA-Cr). For  $2\tilde{K}$ , DA-Cr outperforms the other methods. Interestingly, there is no clear winner between CSPA, NMF, and HAC.

**Experiments with Higher  $K$ .** In partition difference approaches, increasing  $K$  does not necessarily lead to a  $\pi^*$  that has more clusters. Instead,  $K$  serves as an upper bound and new clusters will be used in case they reduce the objective.

To demonstrate how different algorithms handle different  $K$  values, Table 5 shows the consensus criteria and the actual number of clusters in  $\pi^*$  for different values of  $K$  (note that  $\tilde{K} = 3$  in Iris). The results show that the performance of the pairwise similarity methods (CSPA, HAC, DA-Sm) degrades as we increase  $K$ . This is associated with the fact the actual number of clusters in  $\pi^*$  is equal to  $K$  which is significantly higher compared to the clusterings in  $\Pi$ . Methods based on partition difference (NMF and DA-Cr) do not exhibit significant degradation and the actual number of clusters does not grow beyond 5 for DA-Cr and 6 for NMF. Note that the average number of clusters in  $\Pi$  is 5.26.

**Table 4.** Clustering accuracy measured by ARI compared to true labels

Dataset	$\tilde{K}$ clusters					$2\tilde{K}$ clusters				
	CSPA	NMF	HAC	DA-Sm	DA-Cr	CSPA	NMF	HAC	DA-Sm	DA-Cr
Iris	<b>0.868</b>	0.746	0.746	0.716	0.730	0.438	0.463	0.447	0.433	<b>0.521</b>
MNIST	0.684	0.518	0.704	<b>0.730</b>	0.720	0.412	0.484	<b>0.545</b>	0.440	0.484
Optdig.	0.712	0.642	0.675	0.734	<b>0.738</b>	0.380	0.513	<b>0.630</b>	0.481	0.623
Pendig.	0.674	<b>0.679</b>	0.499	0.668	0.668	0.398	0.614	0.625	0.490	<b>0.639</b>
Protein	0.365	0.298	0.363	0.349	<b>0.376</b>	0.237	0.332	0.301	0.308	<b>0.345</b>
Seeds	0.705	0.710	0.704	<b>0.764</b>	0.717	0.424	0.583	0.573	0.500	<b>0.619</b>
Wine	0.324	0.395	0.371	<b>0.402</b>	0.398	0.231	0.245	0.240	<b>0.248</b>	0.238
# Best	1	1	0	<b>3</b>	2	0	0	2	1	<b>4</b>

**Table 5.** Results for Iris dataset with different number of clusters

$K$	Consensus Criteria					# of clusters in consensus clustering				
	CSPA	NMF	HAC	DA-Sm	DA-Cr	CSPA	NMF	HAC	DA-Sm	DA-Cr
3	0.555	<b>0.618</b>	<b>0.618</b>	<b>0.619</b>	<b>0.621</b>	3	3	3	3	3
6	0.536	0.614	0.627	0.608	<b>0.642</b>	6	6	6	6	5
9	0.447	0.614	0.591	0.497	<b>0.642</b>	9	6	9	9	5
12	0.370	0.614	0.507	0.414	<b>0.642</b>	12	6	12	12	5

## 5 Conclusion

Motivated by the recent emergence of specialized hardware platforms, we present a new approach to the consensus clustering problem that is based on Ising models and solved on the Fujitsu Digital Annealer, a specialized CMOS hardware. We perform an extensive empirical evaluation and show that our approach outperforms existing methods on a set of seven datasets. These results show that using specialized hardware in core data mining tasks can be a promising research direction. As future work, we plan to investigate additional problems in data mining that can benefit from the use of specialized optimization hardware as well as experimenting with different types of specialized hardware platforms.

## References

1. Aramon, M., Rosenberg, G., Valiante, E., Miyazawa, T., Tamura, H., Katzgraber, H.G.: Physics-inspired optimization for quadratic unconstrained problems using a digital annealer. *Front. Phys.* **7**, 48 (2019)
2. Bian, Z., Chudak, F., Macready, W.G., Rose, G.: The Ising model: teaching an old problem new tricks. *D-Wave Syst.* **2** (2010)

3. Coffrin, C., Nagarajan, H., Bent, R.: Evaluating Ising processing units with integer programming. In: Rousseau, L.-M., Stergiou, K. (eds.) CPAIOR 2019. LNCS, vol. 11494, pp. 163–181. Springer, Cham (2019). [https://doi.org/10.1007/978-3-030-19212-9\\_11](https://doi.org/10.1007/978-3-030-19212-9_11)
4. DeGroot, M.H., Schervish, M.J.: Probability and Statistics. Pearson, London (2012)
5. Dua, D., Graff, C.: UCI machine learning repository (2017). <http://archive.ics.uci.edu/ml>
6. Filkov, V., Skiena, S.: Integrating microarray data by consensus clustering. *Int. J. Artif. Intell. Tools* **13**(04), 863–880 (2004)
7. Fred, A.L., Jain, A.K.: Combining multiple clusterings using evidence accumulation. *IEEE TPAMI* **27**(6), 835–850 (2005)
8. Fujitsu: Digital annealer. <https://www.fujitsu.com/jp/digitalannealer/>
9. Ghosh, J., Acharya, A.: Cluster ensembles. *Wiley Interdisc. Rev.: Data Mining Knowl. Discov.* **1**(4), 305–315 (2011)
10. Gionis, A., Mannila, H., Tsaparas, P.: Clustering aggregation. *TKDD* **1**(1), 4 (2007)
11. Hubert, L., Arabie, P.: Comparing partitions. *J. Classif.* **2**(1), 193–218 (1985)
12. Karypis, G., Kumar, V.: Multilevelk-way partitioning scheme for irregular graphs. *J. Parallel Distrib. Comput.* **48**(1), 96–129 (1998)
13. Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P.: Optimization by simulated annealing. *Science* **220**(4598), 671–680 (1983)
14. Kumar, V., Bass, G., Tomlin, C., Dulny, J.: Quantum annealing for combinatorial clustering. *Quantum Inf. Process.* **17**(2), 1–14 (2018). <https://doi.org/10.1007/s11128-017-1809-2>
15. Li, T., Ding, C., Jordan, M.I.: Solving consensus and semi-supervised clustering problems using nonnegative matrix factorization. In: *ICDM*, pp. 577–582 (2007)
16. Li, T., Ogihara, M., Ma, S.: On combining multiple clusterings: an overview and a new perspective. *Appl. Intell.* **33**(2), 207–219 (2010)
17. Lucas, A.: Ising formulations of many np problems. *Front. Phys.* **2**, 5 (2014)
18. Naghsh, Z., Javad-Kalbasi, M., Valaee, S.: Digitally annealed solution for the maximum clique problem with critical application in cellular v2x. In: *ICC*, pp. 1–7 (2019)
19. Negre, C.F.A., Ushijima-Mwesigwa, H., Mniszewski, S.M.: Detecting multiple communities using quantum annealing on the d-wave system. *PLoS ONE* **15**, 1–14 (2020)
20. Nguyen, N., Caruana, R.: Consensus clusterings. In: *ICDM*, pp. 607–612 (2007)
21. Rahman, M.T., Han, S., Tadayon, N., Valaee, S.: Ising model formulation of outlier rejection, with application in WiFi based positioning. In: *ICASSP*, pp. 4405–4409 (2019)
22. Rousseeuw, P.J.: Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.* **20**, 53–65 (1987)
23. Shaydulin, R., Ushijima-Mwesigwa, H., Safro, I., Mniszewski, S., Alexeev, Y.: Network community detection on small quantum computers. *Adv. Quantum Technol.* **2**, 1900029 (2019)
24. Strehl, A., Ghosh, J.: Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *JMLR* **3**(Dec), 583–617 (2002)
25. Topchy, A., Jain, A.K., Punch, W.: Clustering ensembles: models of consensus and weak partitions. *IEEE TPAMI* **27**(12), 1866–1881 (2005)
26. Ushijima-Mwesigwa, H., Negre, C.F., Mniszewski, S.M.: Graph partitioning using quantum annealing on the d-wave system. In: *PMES*, pp. 22–29 (2017)

27. Vega-Pons, S., Ruiz-Shulcloper, J.: A survey of clustering ensemble algorithms. *IJPRAI* **25**(03), 337–372 (2011)
28. Wu, J., Liu, H., Xiong, H., Cao, J., Chen, J.: K-means-based consensus clustering: a unified view. *IEEE TKDE* **27**(1), 155–169 (2014)
29. Xing, E.P., Jordan, M.I., Russell, S.J., Ng, A.Y.: Distance metric learning with application to clustering with side-information. In: *NIPS*, pp. 521–528 (2003)

**Open Access** This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

